IN THE CLAIMS:
Please revise the claims, as follows:

1. (Currently amended) A microprocessor system for executing instructions described in a program, said system comprising:

a main processor for executing, by hardware, instructions which belong to a first instruction set and for executing, by software, instructions which belong to a second instruction set, said main processor including an interrupt request reception circuit to decode an interrupt vector for said execution of an instruction of said second instruction set by using an interrupt handler; and

a co-processor operative under the control of said main processor for autonomously fetching an instruction belonging to said second instruction set to execute same by ~~its~~ hardware of said co-processor, said co-processor including an interrupt request generation circuit for encoding said interrupt vector, said interrupt request generation circuit being connected to said interrupt request reception circuit by at least one signal line and allowing an interrupt vector address to be identified in said main processor.

2. (Original claim) A microprocessor system according to claim 1, wherein said co-processor detects an encounter with a specific one of the instructions belonging to said second instruction set which said co-processor cannot process by itself and issues a notification of said encounter to said main processor to thereby request the main processor to execute said specific instruction.

3. (Previously presented) A microprocessor system according to claim 2, wherein said co-processor detects an encounter with a specific one of the instructions belonging to said second instruction set for which data presently under the control of said main processor needs to be handled to thereby determine that said co-processor has encountered a specific instruction which cannot be processed by itself.

4. (Previously presented) A microprocessor system according to claim 2, wherein said co-processor issues said notification by dedicated interrupt vectors assigned in advance, respectively, to a predetermined number of the instructions belonging to said second instruction set which have a higher frequency of execution than the other instructions.

5. (Previously presented) A microprocessor system according to claim 4, wherein at least one of said dedicated interrupt vectors is assigned to a plurality of instructions belonging to said second instruction set.

6. (Previously presented) A microprocessor system according to claim 4, wherein priorities are set to a plurality of said dedicated interrupt vectors.

7. (Original claim) A microprocessor system according to claim 6, wherein a single instruction is assigned to a given one of said dedicated interrupt vectors to which a higher priority is set, while a plurality of instructions are assigned to a given one of said dedicated interrupt vectors to which a lower priority is set.

8. (Previously presented) A microprocessor system according to claim 1, wherein said co-processor further comprises:

a stack memory for holding data generated in the course of execution of an instruction which belongs to said second instruction set;

a stack pointer for holding an address of the most recent data in said stack memory; and

a hardware resource for carrying out a process for updating said stack pointer among processes which take place in the course of execution of said specific instruction.

9. (Previously presented) A microprocessor system according to claim 2, wherein said co-processor further comprises:

a program counter for holding an address of an instruction which is currently processed and belongs to said second instruction set; and

a hardware resource for carrying out a process for updating said program counter among processes which take place in the course of execution of said specific instruction.

10. (Previously presented) A microprocessor system according to claim 2, wherein said co-processor further comprises:

a status register for holding information indicative of a need of said notification, and wherein said main processor periodically accesses said status register to recognize, from a content of said status register, that said co-processor has encountered said specific instruction to thereby execute said specific instruction.

11. (Previously presented) A microprocessor system according to claim 4, wherein said interrupt request reception circuit in said main processor encodes said dedicated interrupt vectors sent from said co-processor to specify an interrupt handler which corresponds to said specific instruction to be processed.

12. (Original claim) A microprocessor system according to claim 2, wherein said co-processor further comprises an instruction queue for holding a fetched instruction which belongs to said second instruction set and wherein said main processor refers to said instruction queue of said co-processor to specify an interrupt handler which corresponds to said specific instruction to be executed.

13. (Previously presented) A microprocessor system according to claim 1, wherein said co-processor includes a stack architecture.

14. (Previously presented) A microprocessor system according to claim 13, further comprising:

a stack memory provided outside said co-processor,

wherein said co-processor further comprises:

a stack-top register for holding a predetermined number of top data of stack data.

15. (Previously presented) A microprocessor system according to claim 14, wherein said co-processor further comprises:

a cache memory provided between said stack memory and said stack-top register for

caching a part of data held in said stack memory.

16. (Currently amended) A microprocessor system according to claim 14, wherein said co-processor detects a predetermined instruction for which said stack data needs to be manipulated over said stack-top register and said stack memory, whereupon said co-processor moves contents of said stack-top register to said stack memory and thereafter requests said main processor to execute said predetermined instruction, said main processor referring to contents of said stack memory, to which said contents of said stack-top register have been moved, to thereby execute said predetermined instruction.

17. (Previously presented) A microprocessor system according to claim 1, further comprising:

a plurality of co-processors in correspondence with a plurality of processes described in a program.

18. (Previously presented) A microprocessor system according to claim 1, further comprising:

a program memory in which instructions belonging to said second instruction set are contained, wherein said co-processor further comprises:

a program counter for holding an address of an instruction that is currently processed and belongs to said second instruction set;

an instruction queue for holding instructions which belong to said second instruction set; and

an instruction fetch circuit for fetching an instruction belonging to said second instruction set from said program memory using a value contained in said program counter as its address and for setting the fetched instruction to said instruction queue.

19. (Currently amended) A method of processing computer programs, said method comprising:

using a main processor for executing hardware instructions which belong to a first instruction set and for executing software instructions which belong to a second instruction set;

using a co-processor, operative under the control of said main processor, for autonomously fetching instructions belonging to said second instruction set to execute the fetched instuctions by ~~its~~ hardware of said co-processor, wherein said co-processor is unable to execute at least one instruction in said second instruction set; and

generating an interrupt request from said co-processor to said main processor when said co-processor detects encountering said at least one ~~of the instructions~~ instruction belonging to said second instruction set which said co-processor cannot ~~process~~ execute by itself, thereby requesting that said main processor execute said instruction,

wherein said interrupt request comprises a signal on at least one signal line between said main processor and said co-processor dedicated to an interrupt vector signal, said interrupt vector comprising a dedicated interrupt vector component and a common interrupt request component, said dedicated interrupt vector component comprising an encoding for a specific interrupt handler to be executed by said main processor and said common interrupt request component providing an indication for a request for one of a plurality of other

interrupt handlers to be specifically identified by additional information.

20. (Previously presented) The method of claim 19, further comprising:

converting said encoded signal into an interrupt handler address in said main processor.

21. (Previously presented) The method of claim 19, wherein said at least one signal line comprises a plurality of signal lines.

22. (Currently amended) A microprocessor system, comprising:

a main processor including an interrupt request reception circuit to decode an interrupt vector, said interrupt vector comprising a dedicated interrupt vector component and a common interrupt request component;

a co-processor operative under the control of said main processor for autonomously fetching and executing an instruction, said co-processor including an interrupt request generation circuit for encoding said interrupt vector; and

at least one signal line interconnecting said interrupt request generation circuit and said interrupt request reception circuit,

wherein said dedicated interrupt vector component comprises an encoding for a specific interrupt handler to be executed by said main processor and said common interrupt request component provides an indication for a request for one of a plurality of interrupt handlers to be specifically identified by additional information.

23. (Previously presented) The microprocessor system of claim 22, wherein said at least one signal line comprises a plurality of signal lines.

24. (New) A microprocessor system for executing instructions described in a program, comprising:

a main processor for executing by hardware those instructions which belong to a first instruction set and for executing by software those instructions which belong to a second instruction set; and

a co-processor operative under the control of said main processor for autonomously fetching an instruction belonging to said second instruction set to execute the fetched instruction by hardware of the co-processor,

wherein said coprocessor is provided with:

a stack memory for holding data generated in the course of execution of an instruction which belongs to said second instruction set;

a stack pointer for holding an address of the most recent data in said stack memory;

a program counter for holding an address of an instruction which is currently processed and belongs to said second instruction set; and

an updating circuit for, in response to the detection of an encounter with a specific instruction among instructions belonging to said second instruction set for which data presently under the control of said main processor needs to be handled, issuing a notification of said encounter to said main processor to request the main processor to execute said specific instruction, and for updating said stack pointer in said stack memory and said program counter.

25. (New) A microprocessor system according to claim 24, wherein said co-processor issues said notification by dedicated interrupt vectors assigned in advance respectively to a predetermined number of instructions among the instructions belonging to said second instruction set which have a higher frequency of execution than the other instructions.

26. (New) A microprocessor system according to claim 24, wherein said co-processor comprises a status register for holding information indicative of a need of said notification, and

wherein said main processor periodically accesses said status register to recognize that said co-processor has encountered said specific instruction based on said status register, to thereby execute said specific instruction.

27. (New) A microprocessor system according to claim 25, wherein said main processor comprises an interrupt request reception circuit for encoding said dedicated interrupt vectors sent from said co-processor to specify an interrupt handler which corresponds to said specific instruction to be processed.

28. (New) A microprocessor system according to claim 24, wherein said main processor refers to an instruction queue of said co-processor to specify an interrupt handler which corresponds to said specific instruction to be executed.

29. (New) A microprocessor system according to claim 24, wherein said co-processor includes a stack architecture.

30. (New) A microprocessor system according to claim 25, wherein said co-processor includes a stack architecture.

31. (New) A microprocessor system according to claim 29, wherein said co-processor comprises:

a stack-top register for holding a predetermined number of top data of stack data; and

a cache memory provided between said stack memory provided outside said co-processor and said stack-top register for caching a part of data held in said stack memory.

32. (New) A microprocessor system according to claim 30, wherein said co-processor comprises:

a stack-top register for holding a predetermined number of top data of stack data; and

a cache memory provided between said stack memory provided outside said co-processor and said stack-top register for caching a part of data held in said stack memory.

33. (New) A microprocessor system according to claim 24, further comprising a plurality of co-processors corresponding to a plurality of processes described in said program.

34. (New) A microprocessor system according to claim 25, further comprising a plurality of co-processors corresponding to a plurality of processes described in said program.

35. (New) A microprocessor system according to claim 26, further comprising a plurality of co-processors corresponding to a plurality of processes described in said program.